

CLAIMS:

- Sub Sai
1. A method of generating a source code listing from an object code sequence comprising section data including a plurality of program instructions, said section data having associated therewith a relocation section including at least one relocation instruction which is used at link time to modify the object code sequence to generate an executable program, the method comprising:
 - for each location in the section data determining if that location in said section data has a relocation instruction associated with it;
 - reading said associated relocation instruction and deriving from the relocation instruction additional information concerning said section data; and
 - generating the source code listing for that location in the section data, the source code listing comprising the source code from which the object code at that location was derived with said additional information derived from the relocation instruction.
 2. A method according to claim 1, comprising displaying the source code listing in a humanly readable form.
 3. A method according to claim 1, wherein said object code sequence forms part of an object code module.
 4. A method according to claim 1, wherein said code sequence forms part of an executable program.
 5. A method according to claim 1, wherein the object code sequence forms part of a library object code module holding a number of frequently used code sequences together with their associated relocations.

Sub
Part

6. A method according to claim 1, wherein said relocation section comprises a plurality of relocation instructions for performing calculations using a stack.

7. A method according to claim 6, wherein said relocation instructions are read sequentially to derive said additional information, the additional information comprising an arithmetic expression which was defined in the source code but not in the object code sequence.

8. A method according to claim 1, which comprises the step of generating assembler directives in said source code corresponding to relocation instructions in the relocation section.

9. A method according to claim 1, wherein the additional information is operand values defined in the source code but not in the object code sequence.

10. A method according to claim 1, wherein the additional information is event information derived from the relocation instructions.

11. A lister for generating a source code listing from an object code sequence comprising a plurality of program instructions, at least one of said program instructions having a relocation instructions associated with it, the lister comprising:

an instruction reader for reading each said program instruction;

a relocation reader for reading said relocation instructions;

a relocation identifier for determining for each program instruction whether there is an associated relocation instruction; and

a disassembler module for disassembling said program instructions received from said instruction reader to generate source code and for disassembling additional information received from said relocation instruction

Sub
wherein said source code and said additional information can be displayed in human readable form.

12. A lister according to claim 11, further comprising a program count detector for detecting the program count value for each program instruction and offset determining means for determining an offset identified in each relocation instruction to determine if the relocation instruction is associated with the program instruction.

13. A lister according to claim 11, wherein said relocation reader is operable to determine the type of relocation instruction being read.

14. A lister according to claim 13, wherein the types of relocation include an operand relocation; an operator relocation; an event relocation; and a directive relocation.

15. A lister according to claim 11 which comprises an expression calculator for calculating expressions generated from the relocation instructions.

16. A lister according to claim 15, which comprises an expression stack for holding said calculated expressions.

17. A lister according to claim 11, which comprises an event calculator for identifying events associated with an instruction from said relocation instructions.

18. A lister according to claim 17, which comprises an event stack for holding said identified events.

19. A lister according to claim 11, which comprises a directive processor which generates assembler directives derived from the relocation instructions.

*add
a1*